AR-B6050 Board

Fan-less with Intel ATOM Pineview + ICH8M

User Manual

Manual Rev.: 1.0
Book Number: AR-B6050-2010.07.27

Revision

Version	Date	Author	Description
1.0	2010/07/27	Cody Dai	Draft

Copyright 2010

All Rights Reserved.

Manual's first edition:

For the purpose of improving reliability, design and function, the information in this document is subject to change without prior notice and does not represent a commitment on the part of the manufacturer.

In no event will the manufacturer be liable for direct, indirect, special, incidental, or consequential damages arising out of the use or inability to use the product or documentation, even if advised of the possibility of such damages.

This document contains proprietary information protected by copyright. All rights are reserved. No part of this Manual may be reproduced by any mechanical, electronic, or other means in any form without prior written permission of the manufacturer.

Trademarks

AR-B6050 is a registered trademarks of Acrosser; IBM PC is a registered trademark of the International Business Machines Corporation; Pentium is a registered trademark of Intel Technologies Inc; Award is a registered trademark of Award Software International Inc; other product names mentioned herein are used for identification purposes only and may be trademarks and/or registered trademarks of their respective companies.

Table of Contents

1 Introduction 5

1.1 Specifications 6
1.2 Package Contents 7
1.3 Block Diagram 8

2 H/W Information 9

2.1 Locations of Connector and Jumper Setting 9
2.2 Connector and Jumper Setting Table 12

3 BIOS Setting 16

3.1 Main Setup 17
3.2 Advanced Chipset Setup 19
3.3 Peripherals Setup 20
3.4 Power Setup 22
3.5 Boot Setup 23
3.6 Exit Setup 24

4 WATCHDOG, GPIO, AND BYPASS PROGRAMMING 25

4.1 Watchdog Programming 25
4.2 GPIO Programming 29

1 INTRODUCTION

AR-B6050 is a 3.5” SBC board that is designed with Intel Atom N450 and supports up to 2GB of DDR2 667Mhz memory. AR-B6050 has diverse physical interface for different peripheral, e.g. VGA port, LVDS port, 6 * USB 2.0 ports, 2 * COM ports, 2 * Gbps ports, 2 * SATA ports, CF type I/II slot and Realtek audio output port. It is also equipped with industrial standard PCI-104 and miniPCIe interface. Users can purchase suitable add-on cards to satisfy their needs.

1.1 Specifications

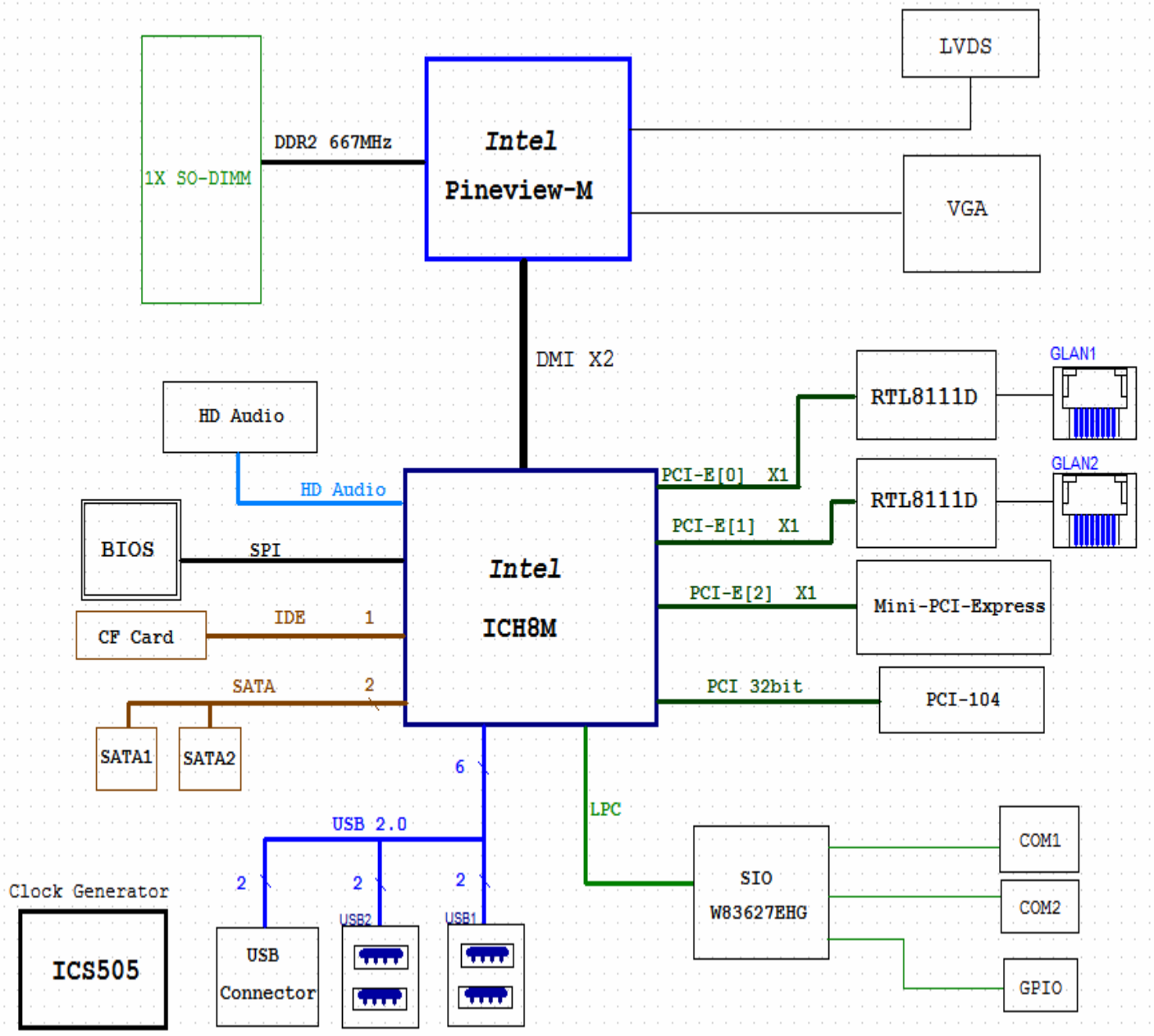
- IntelR Atom N450 1.66GHz
- IntelR Graphics Media Accelerator 950
- 1 x SO-DIMM supports DDRII up to 2GB(Memory DDR2 data transfer rates of 667 MT/s)
- 1 x VGA
- 4 x USB2.0
- 2 x SATA
- 1 x CF II
- 2 x RS-232
- 2 x GbE (Realtek RTL8111D)
- 1 x PCI-104 & 1 x Mini-PCIe
- 8-bit GPIO

1.2 Package Contents

Check if the following items are included in the package.

- Quick Manual
- AR-B6050
- 1 x Software Utility CD

1.3 Block Diagram

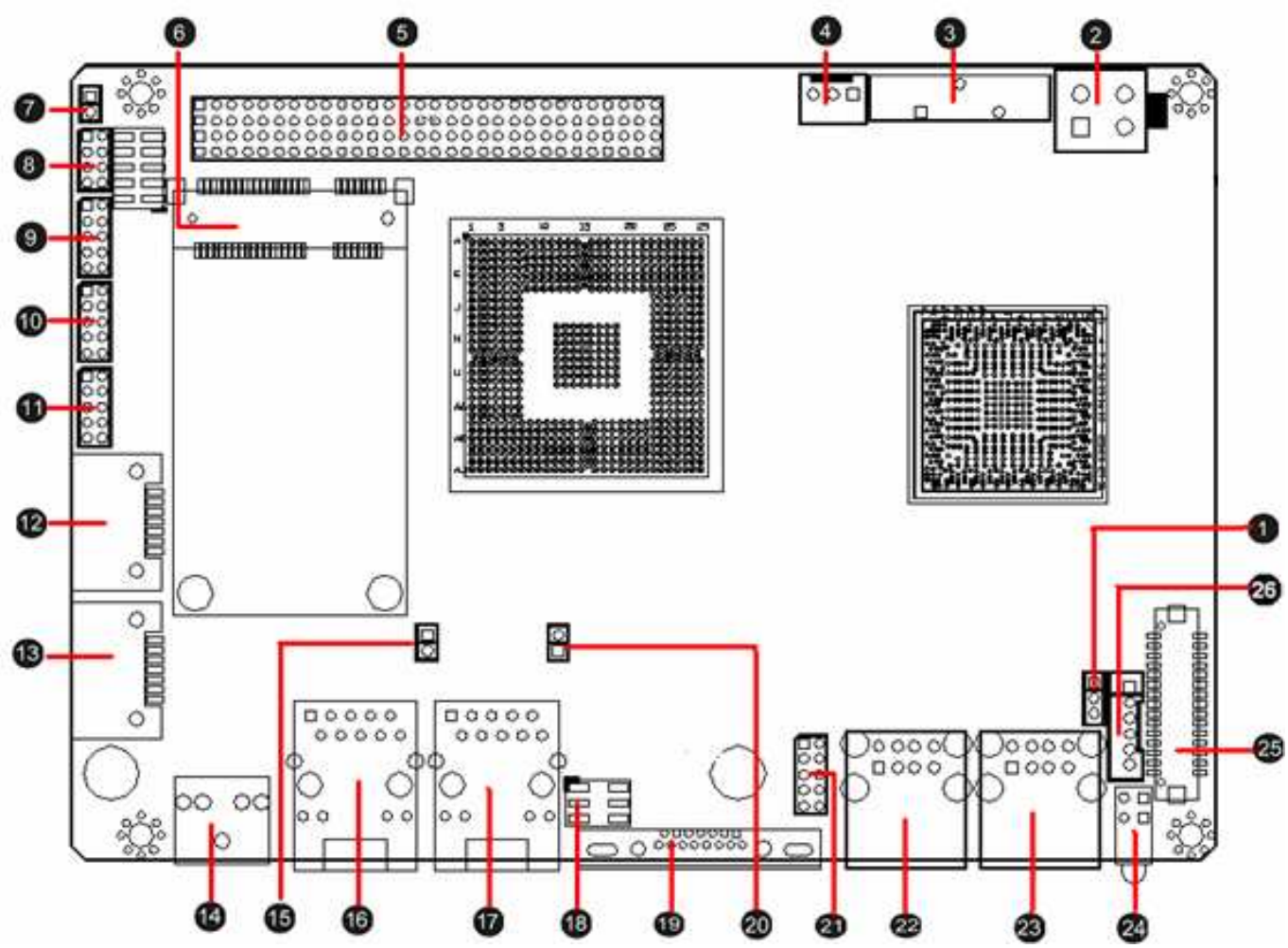


2 H/W INFORMATION

This chapter describes the installation of AR-B6050. At first, it shows the Function diagram and the layout of AR-B6050. It then describes the unpacking information which you should read carefully, as well as the jumper/switch settings for the AR-B6050 configuration

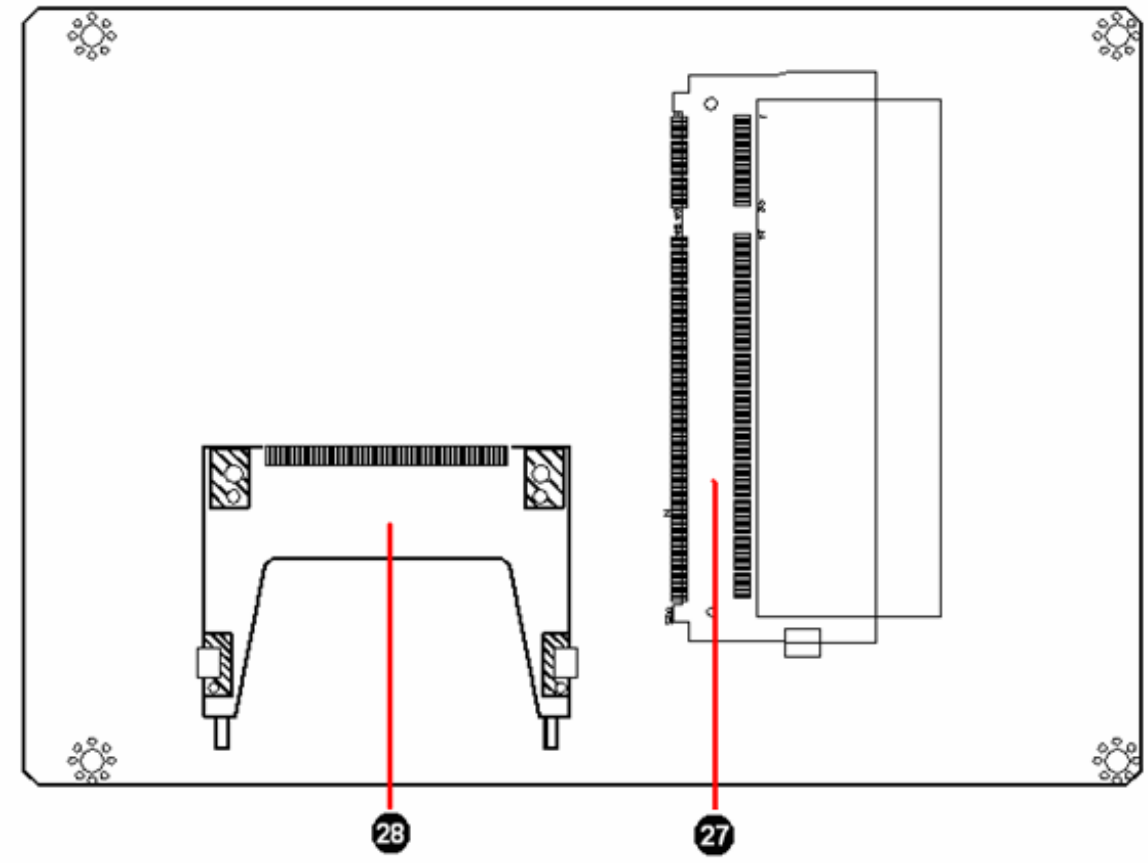
2.1 Locations of Connector and Jumper Setting

2.1.1 Locations (Top side)



1	JP1	10	COM1	19	VGA1
2	ATX1	11	COM2	20	J1
3	BAT1	12	SATA2	21	USB3
4	SYSFAN1	13	SATA1	22	USB1
5	CN2	14	CN4	23	USB2
6	CN3	15	JP3	24	LED1
7	JP2	16	LAN1	25	LVDS1
8	J7	17	LAN2	26	CN1
9	GPIO1	18	J6		

2.1.2 Locations (Bottom Side)



27	SODIMM1
28	CF

2.2 Connector and Jumper Setting Table

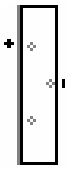
1. JP1: LCD panel driving voltage selection.

STATUS	SETTING
1-2	+3.3V (Default).
2-3	+5V

2. ATX1: AT power input connector.

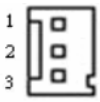
PIN	SETTING
1	GND
2	GND
3	+12V
4	+12V

3. BAT1: CMOS battery holder.



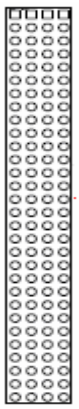
CMOS battery holder.

4. SYSFAN1: System DC Fan connector.



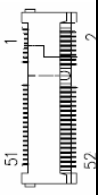
PIN	SETTING
1	GND
2	+12V
3	Fan speed data

5. CN2: PCI-104 connector.



PCI-104 connector.

6. CN3: MINI PCI-E connector.

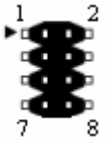


MINI PCI-E connector.

7. JP2: Signal SERIRQ connects to PCI-104 pin #B1 selection.

STATUS	SETTING
Open	Disconnected. (Default)
Short	Connected.

8. J7: COM1/2 SELECT RI OR +12V



PIN	SIGNAL	PIN	SIGNAL
1	RI#1	2	RI#1_12V
3	+12V	4	RI#1_12V
5	RI#2	6	RI#2_12V
7	+12V	8	RI#2_12V

9. GPIO1: GPIO connector.

PIN	SETTING	PIN	SETTING
1	GPIO0	2	+5V
3	GPIO1	4	GPIO7
5	GPIO2	6	GPIO6
7	GPIO3	8	GPIO5
9	GND	10	GPIO4

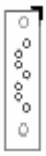
10. COM1: RS232 signal connector for port #1.

PIN	SETTING	PIN	SETTING
1	DCD #1	2	DSR #1
3	RX #1	4	RTS #1
5	TX #1	6	CTS #1
7	DTR #1	8	RI #1
9	GND	10	GND

11. COM2: RS232 signal connector for port #2.

PIN	SETTING	PIN	SETTING
1	DCD #2	2	DSR #2
3	RX #2	4	RTS #2
5	TX #2	6	CTS #2
7	DTR #2	8	RI #2
9	GND	10	GND

12. SATA1: SATA device connector #1.

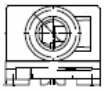


SATA device connector #1.

13. SATA2: SATA device connector #2.

SATA device connector #2.

14. CN4: Audio signal connector.



Audio line out

15. JP3: CF MASTER SELECT

SET	SIGNAL
SHORT	MASTER
OPEN	SLAVE

16. LAN1: RJ45 connector for Gigabit Ethernet port #1.

RJ45 connector for Gigabit Ethern*et* port #1.

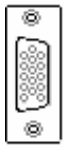
17. LAN2: RJ45 connector for Gigabit Ethernet port #2.

RJ45 connector for Gigabit Ethern*et* port #2.

18. J6: Front panel connector.

STATUS	SETTING
1-2	Hardware reset
3-4	AT Mode - Short ATX Mode - Open
5-6	Power Button

19. VGA1: D-SUB-15 female connector for VGA output.



D-SUB-15 female connector for VGA output.

20. J1: CMOS data clear

SHORT CMOS data clear

21. USB3: Internal USB2.0 connector

PIN	SETTING	PIN	SETTING
1	+5V	2	+5V
3	USB5-	4	USB6-
5	USB5+	6	USB6+
7	GND	8	GND
9	GND	10	GND

22. USB1: USB connector

Upper: Port #2.

Lower: Port #1.

23. USB2: USB connector

Upper: Port #4.

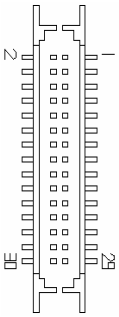
Lower: Port #3.

24. LED1: System power and HDD access indicators.

Green: System power indicator.

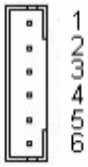
Yellow: HDD access indicator.

25. LVDS1: LCD panel inverter power connector.

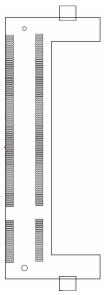


PIN	SETTING	PIN	SETTING
1	LCD VDD	2	GND
3	NC	4	NC
5	GND	6	NC
7	NC	8	GND
9	NC	10	NC
11	NC	12	NC
13	NC	14	NC
15	GND	16	O CLK+
17	CLK-	18	GND
19	Data2+	20	Data2-
21	I2C CLK	22	Data1+
23	Data1-	24	I2C Data
25	Data0+	26	Data0-
27	NC	28	NC
29	LCD VDD	30	LCD VDD

26. CN1: LCD panel inverter power connector.



PIN	SETTING
1	+12V
2	+12V
3	GND
4	BKL ON
5	GND
6	Reserved.

27. DIMM1: DDR-II SODIMM Socket.		28. CF1: Type-II compact flash card socket.		
	DDR-II SODIMM Socket.		+3.3V CF card only and UDMA mode supported.	

3 BIOS SETTING

The BIOS Setup Utility is a hardware configuration program built into your computer's BIOS.

To activate the BIOS Utility, press F2 during POST (when "Press <F2> to enter Setup" message is prompted on the bottom of screen).

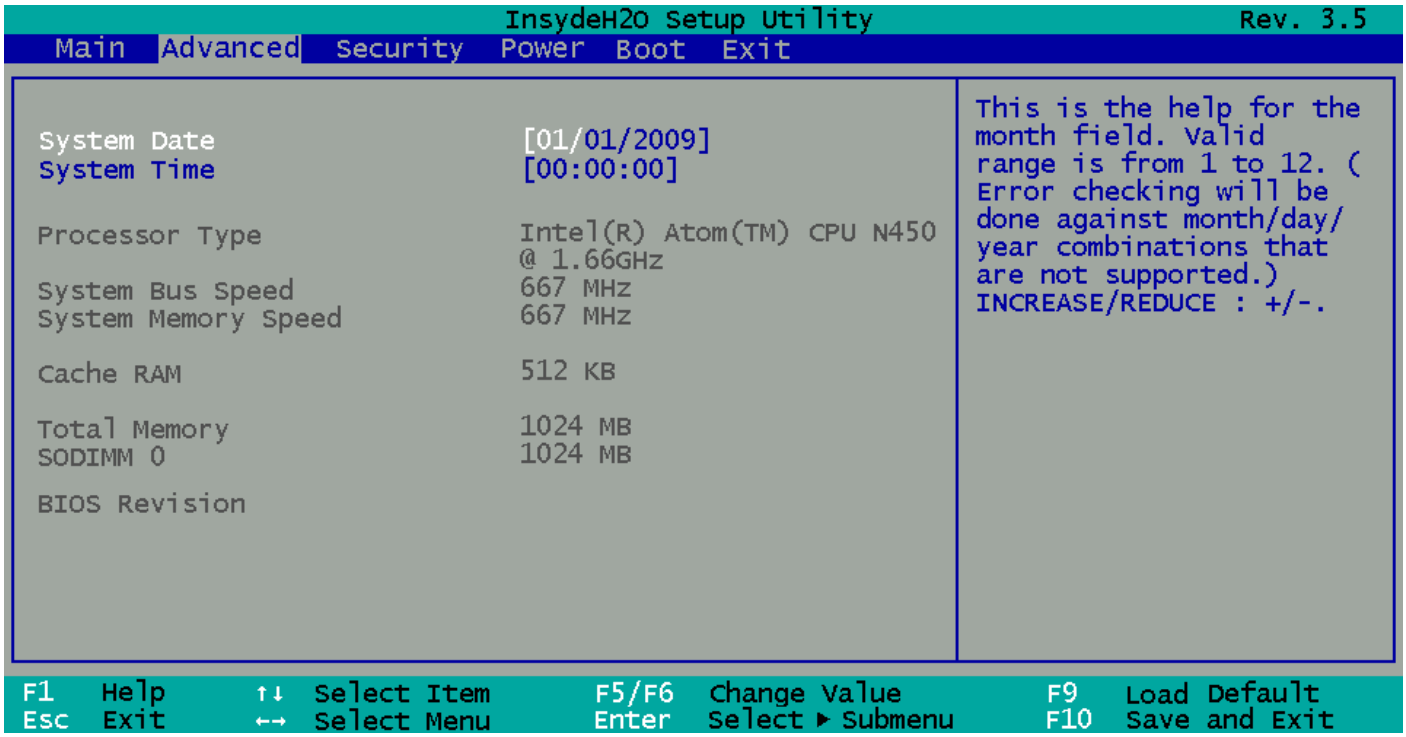
Press <F12> during POST to enter multi-boot menu. In this menu, user can change boot device without entering BIOS SETUP Utility.

This chapter describes the BIOS menu displays and explains how to perform common tasks needed to get the system up and running. It also gives detailed explanation of the elements found in each of the BIOS menu. The following topics are covered:

- Main Setup
- Advanced Setup
- Security Setup
- Power Setup
- Boot Setup
- Exit Setup

3.1 Main Setup

Once you enter the InsydeH2O BIOS™ Setup Utility, the Main Menu will appear on the screen. Use the arrow keys to highlight the item and then use the <F5> <F6> keys to select the desired value in each item.

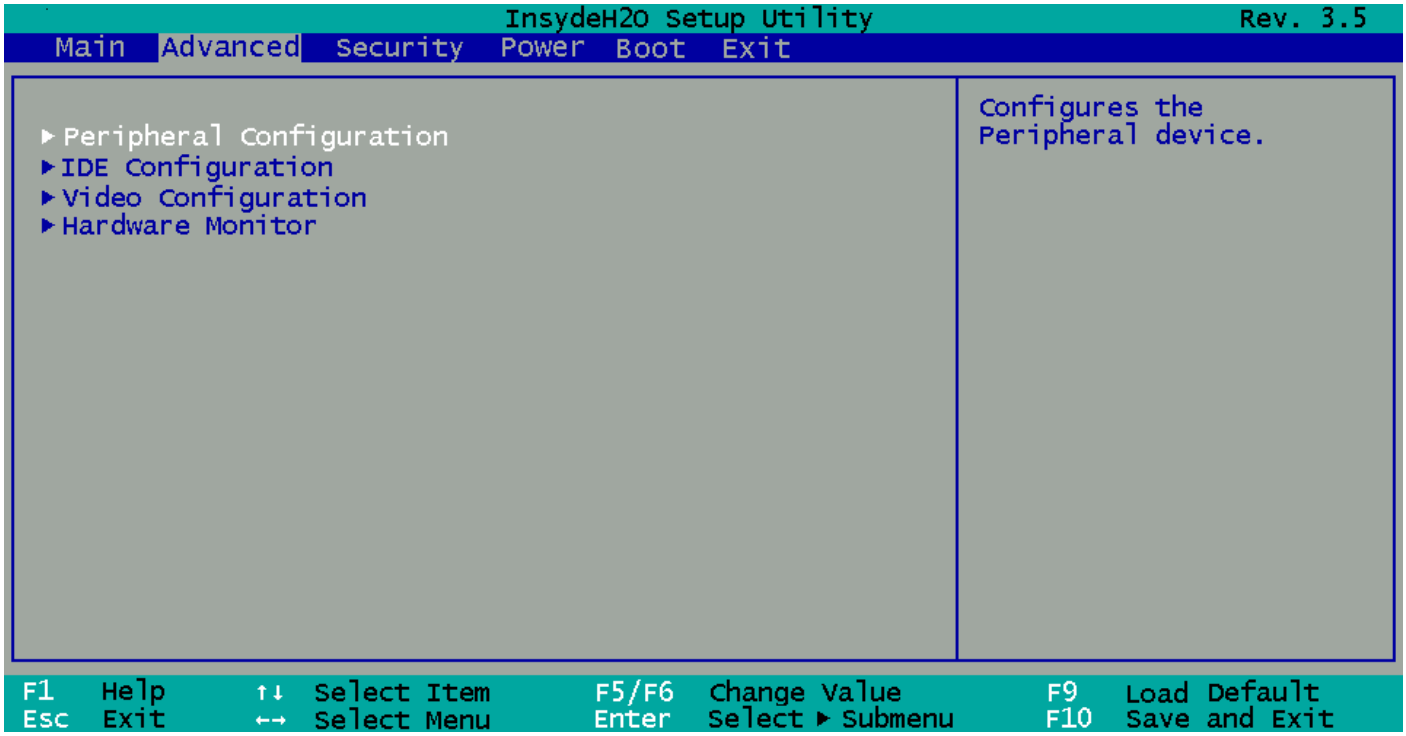

InsydeH2O Setup Utility Rev. 3.5
Main Advanced Security Power Boot Exit
System Date [01/01/2009]
System Time [00:00:00]
Processor Type Intel(R) Atom(TM) CPU N450 @ 1.66GHz
System Bus Speed 667 MHz
System Memory Speed 667 MHz
Cache RAM 512 KB
Total Memory 1024 MB
SODIMM 0 1024 MB
BIOS Revision
This is the help for the month field. Valid range is from 1 to 12. (Error checking will be done against month/day/ year combinations that are not supported.) INCREASE/REDUCE : +/-.
F1 Help ↑↓ Select Item F5/F6 Change Value F9 Load Default
Esc Exit ↔ Select Menu Enter Select ▸ Submenu F10 Save and Exit

Note: The control keys are listed at the bottom of the menu. If you need any help with the item fields, you can press the <F1> key, and the relevant information will be displayed.

Item	Option	Description
System Date	Format : MM/DD/YYYY (month/day/year)	Set the system date. Note that the 'Day' automatically changes when you set the date.
System Time	Format: HH:MM:SS (hour:minute:second)	Set the system time.
Processor Type	N/A	This field shows the CPU type and speed of the system.

System Bus Speed	N/A	This field displays the bus speed of the system.
System Memory Speed	N/A	This field displays the real speed of the memory.
Cache RAM	N/A	This field displays the cache ram of the CPU.
Total Memory	N/A	Displays the total memory available.
SODIMM 0	N/A	This field displays the memory of the SODIMM0.
BIOS Revision	N/A	Displays system BIOS version.

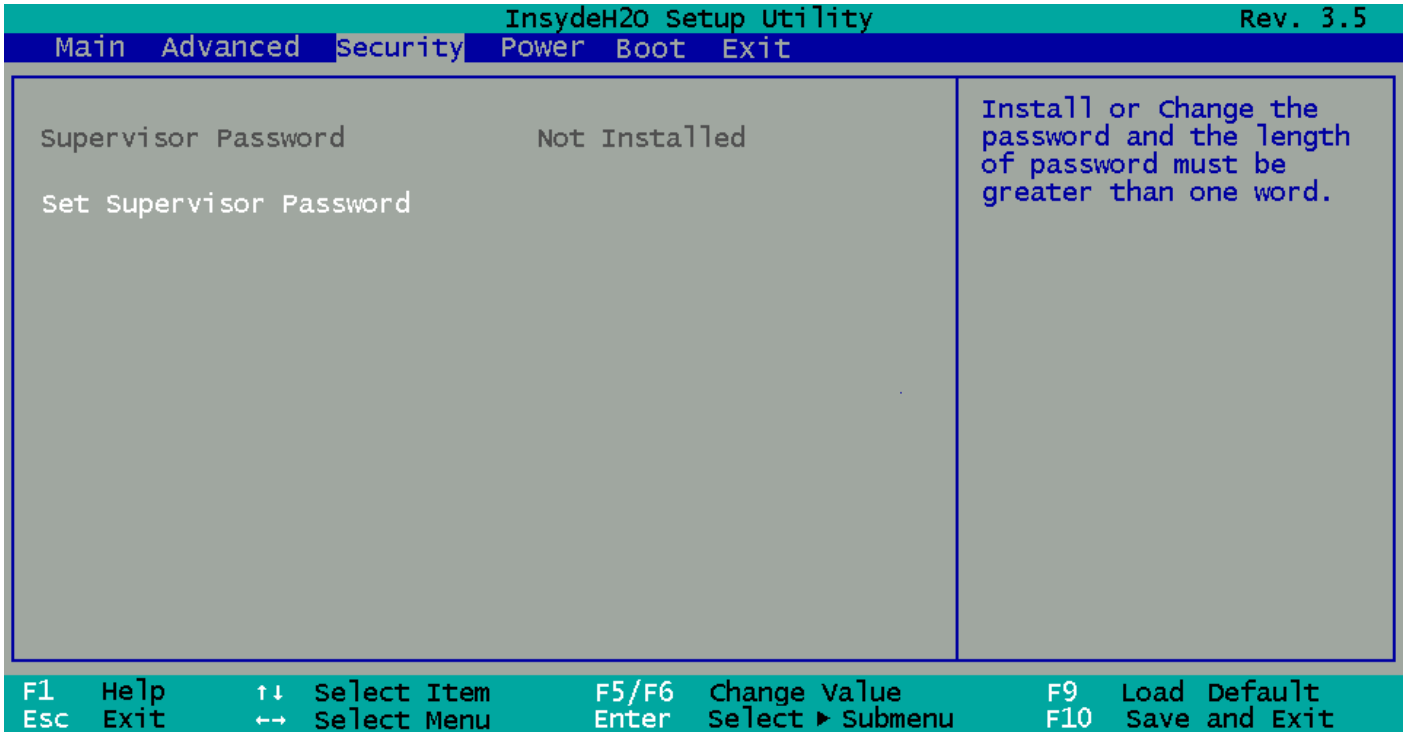
3.2 Advanced Chipset Setup


InsydeH2O Setup Utility Rev. 3.5
Main Advanced Security Power Boot Exit
▶Peripheral Configuration
▶IDE Configuration
▶Video Configuration
▶Hardware Monitor
Configures the Peripheral device.
F1 Help ↑↓ Select Item F5/F6 Change Value F9 Load Default
Esc Exit ↔ Select Menu Enter Select ▶ Submenu F10 Save and Exit

Note: The control keys are listed at the bottom of the menu. If you need any help with the item fields, you can press the <F1> key, and the relevant information will be displayed.

Item	Option	Description
Peripheral Configuration	Serial Port A Serial Port B Azalia Audio control	Enter the Peripheral Configuration menu.
IDE Configuration	IDE Controller HDC Configure as	Enter the IDE Configuration menu.
Video Configuration	Init Display First IGD-Device2, Function1 IGD-Frame Buffer Size IGD-DVMT Size IGD-Boot Type IGD-LCD Panel type	Enter the Video Configuration menu.
Hardware Monitor	N/A	This field displays the Hardware Monitor of the system.

3.3 Peripherals Setup


InsydeH2O Setup Utility Rev. 3.5
Main Advanced Security Power Boot Exit
Supervisor Password Not Installed
Set Supervisor Password
Install or Change the password and the length of password must be greater than one word.
F1 Help ↑↓ Select Item F5/F6 Change Value F9 Load Default
Esc Exit ↔ Select Menu Enter Select ▶ Submenu F10 Save and Exit

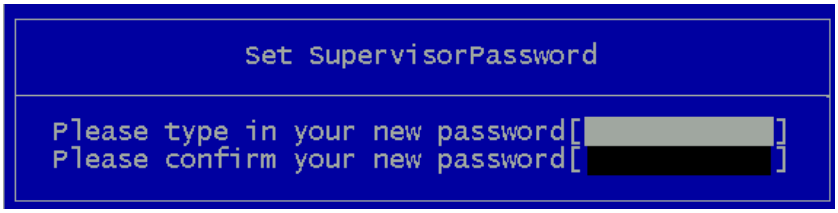
Note: The control keys are listed at the bottom of the menu. If you need any help with the item fields, you can press the <F1> key, and the relevant information will be displayed.

Item	Option	Description
Supervisor Password	Not Installed Installed	Shows the setting of the Supervisor password
Set Supervisor Password	N/A	Press Enter to set the user password. When user password is set, this password protects the BIOS Setup Utility from unauthorized access. The user can enter Setup menu only and does not have right to change the value of parameters.

Setting a Password

Follow these steps as you set the user or the supervisor password:

1. Use the ↑ and ↓ keys to highlight the Set Supervisor Password parameter and press the Enter key. The Set Supervisor Password box appears:



2. Type a password in the “Enter New Password” field. The password length can not exceed 8 alphanumeric characters (A-Z, a-z, 0-9, not case sensitive). Retype the password in the “Confirm New Password” field.

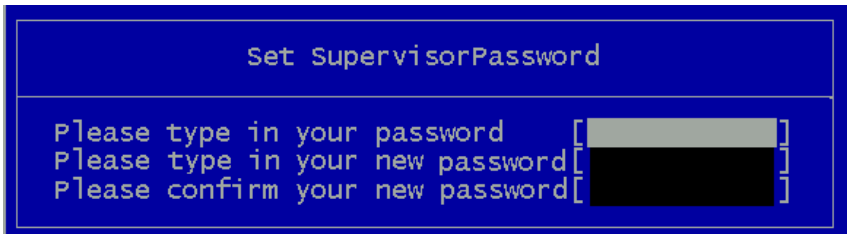
IMPORTANT:Be very careful when typing your password because the characters do not appear on the screen.

3. Press Enter. After setting the password, the computer sets the User Password parameter to “Set”.
4. If desired, you can opt to enable the Password on boot parameter.
5. When you are done, press F10 to save the changes and exit the BIOS Setup Utility.

Removing a Password

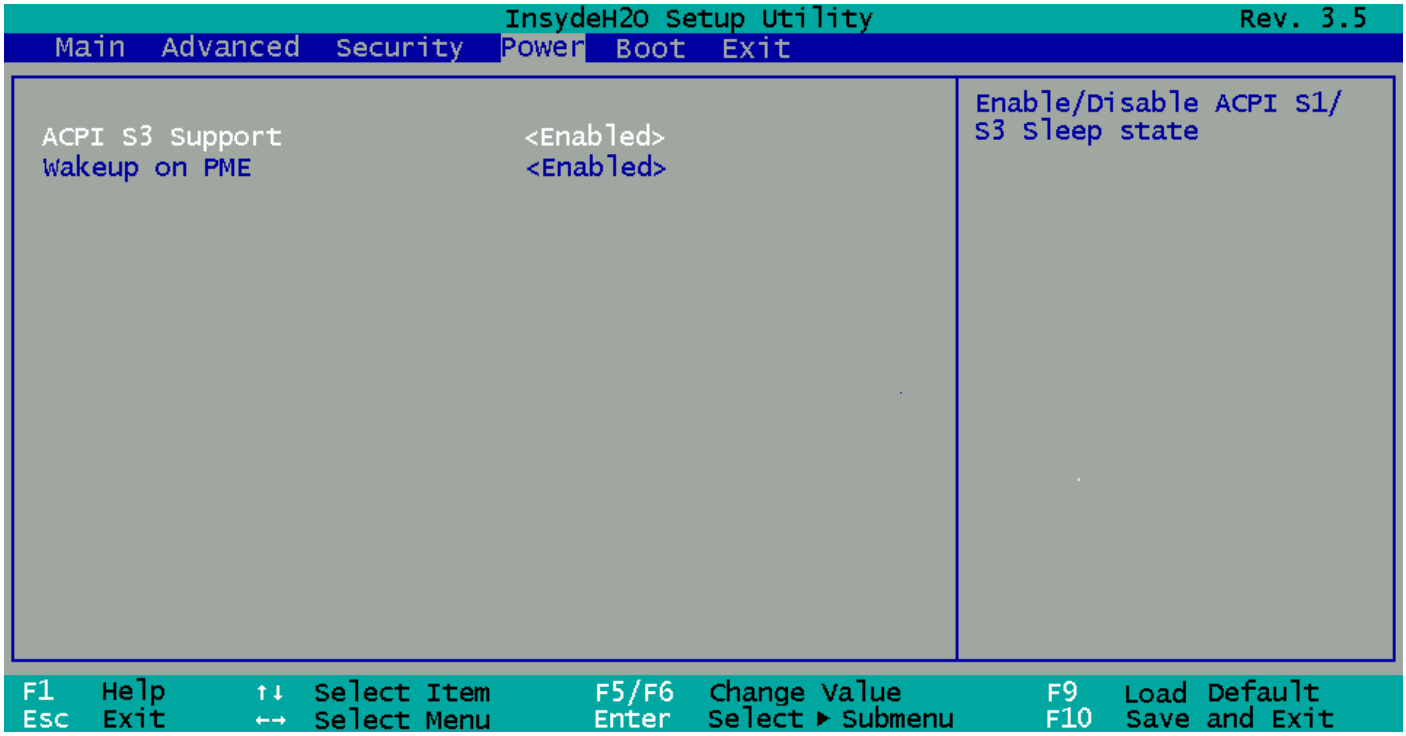
Follow these steps:

1. Use the ↑ and ↓ keys to highlight the Set Supervisor Password parameter and press the Enter key. The Set Password box appears:


Set SupervisorPassword
Please type in your password
Please type in your new password[
Please confirm your new password[

2. Type the current password in the Enter Current Password field and press Enter.
3. Press Enter twice without typing anything in the Enter New Password and Confirm New Password fields. The computer then sets the Supervisor Password parameter to “Clear”.
4. When you have changed the settings, press u to save the changes and exit the BIOS Setup Utility.

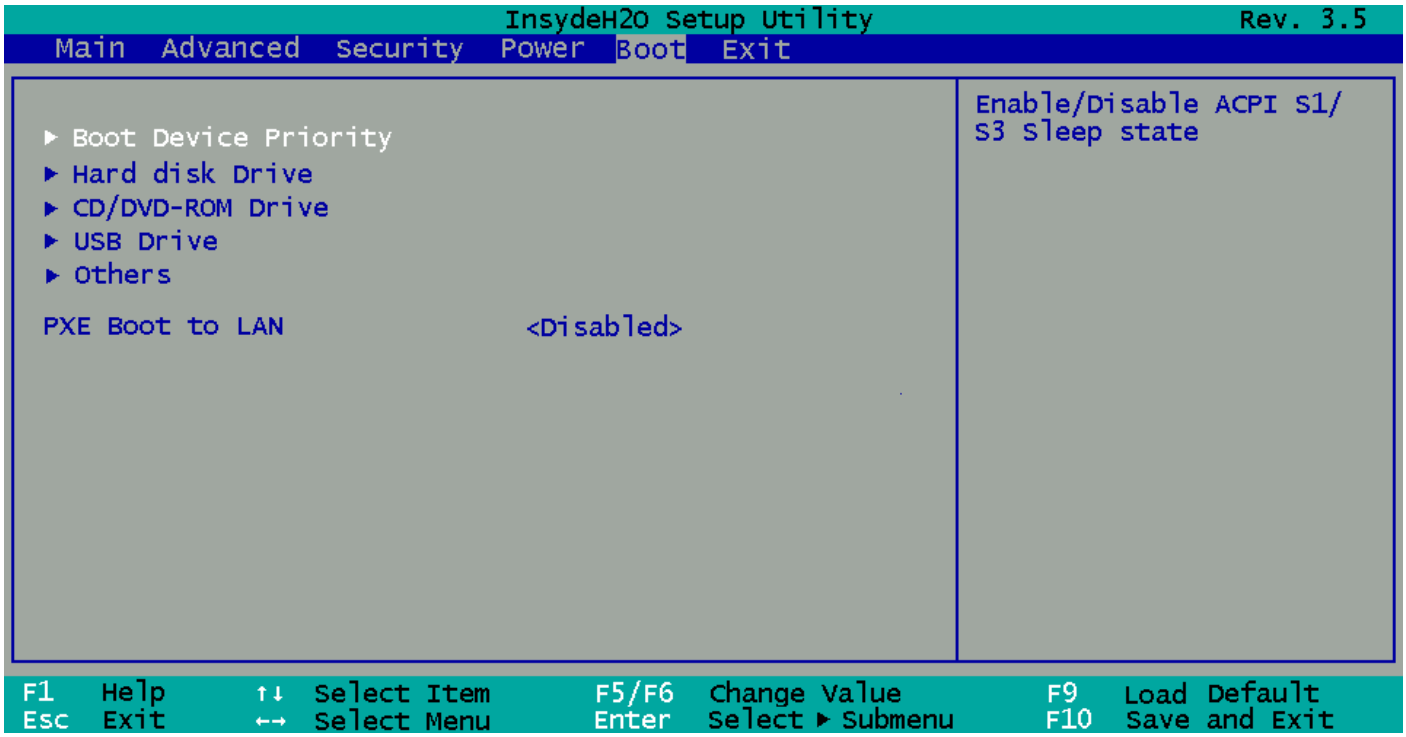
3.4 Power Setup


InsydeH2O Setup Utility Rev. 3.5
Main Advanced Security Power Boot Exit
ACPI S3 Support <Enabled>
Wakeup on PME <Enabled>
Enable/Disable ACPI S1/ S3 Sleep state
F1 Help ↑↓ Select Item F5/F6 Change Value F9 Load Default
Esc Exit ↔ Select Menu Enter Select ▶ Submenu F10 Save and Exit

Note: The control keys are listed at the bottom of the menu. If you need any help with the item fields, you can press the <F1> key, and the relevant information will be displayed.

Item	Option	Description
ACPI S3 Support	Enabled Disabled	ACPI S1/S3 Sleep State.
Wakeup on PME	Enabled Disabled	Wake up when the system power is off and a PCI Power Management Enable wake up event occurs.

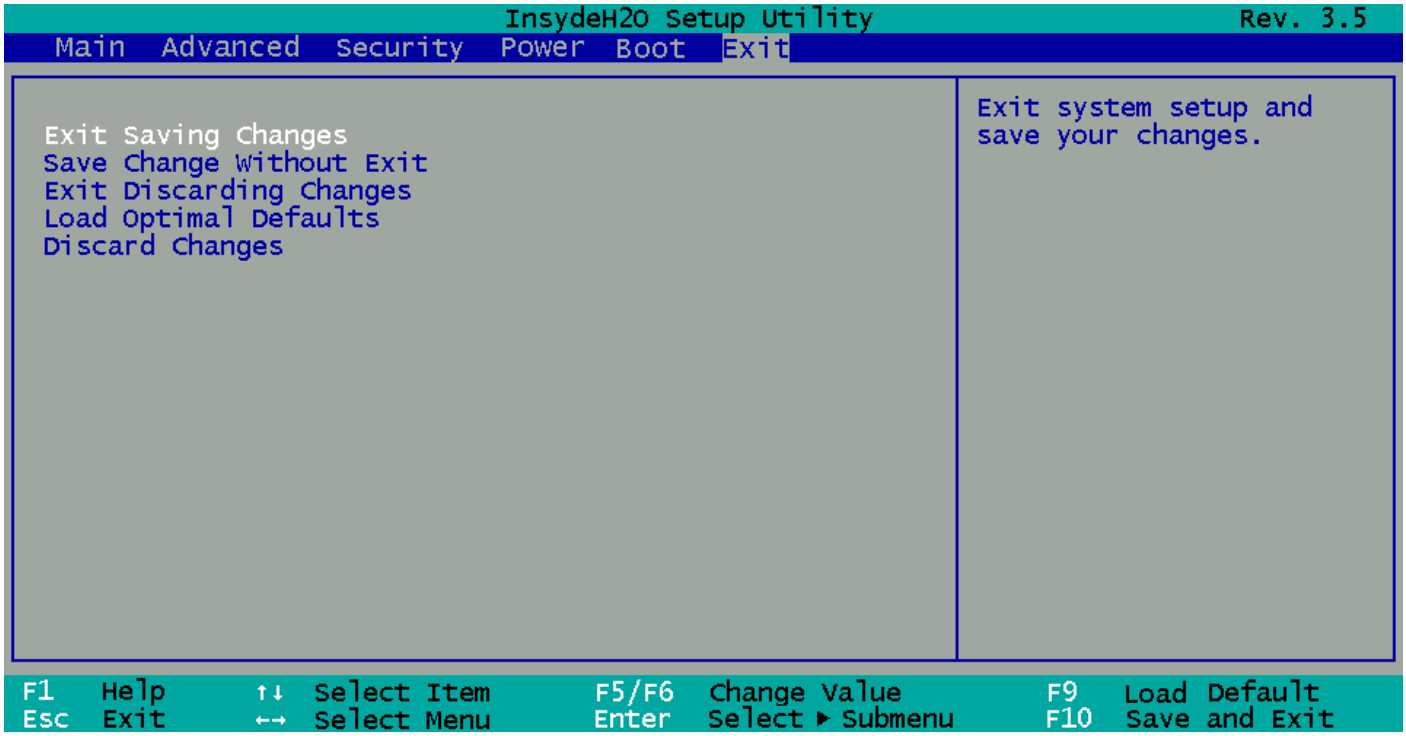
3.5 Boot Setup


InsydeH2O Setup Utility Rev. 3.5
Main Advanced Security Power Boot Exit
Boot Device Priority
Hard disk Drive
CD/DVD-ROM Drive
USB Drive
Others
PXE Boot to LAN <Disabled>
Enable/Disable ACPI S1/ S3 Sleep state
F1 Help ↑↓ Select Item F5/F6 Change Value F9 Load Default
Esc Exit ↔ Select Menu Enter Select ▶ Submenu F10 Save and Exit

Note: The control keys are listed at the bottom of the menu. If you need any help with the item fields, you can press the <F1> key, and the relevant information will be displayed.

Item	Option	Description
Boot Device Priority	N/A	Select Boot Devices to select specific devices to support boot. (The item can't display when no device.)
Hard disk Drive	N/A	Show the Hard disk drives. (The item can't display when no device.)
CD/DVD-ROM Drive	N/A	Show the CD/DVD-ROM drives. (The item can't display when no device.)
USB Drive	N/A	Show the USB diskette drives. (The item can't display when no device.)
Other	N/A	Show the other drives. (The item can't display when no device.)
PXE Boot to LAN	Enabled Disabled	Disables or enables PXE boot to LAN.

3.6 Exit Setup


InsydeH2O Setup Utility Rev. 3.5
Main Advanced Security Power Boot Exit
Exit Saving Changes
Save Change Without Exit
Exit Discarding Changes
Load Optimal Defaults
Discard Changes
Exit system setup and save your changes.
F1 Help ↑↓ Select Item F5/F6 Change Value F9 Load Default
Esc Exit ↔ Select Menu Enter Select ▶ Submenu F10 Save and Exit

Note: The control keys are listed at the bottom of the menu. If you need any help with the item fields, you can press the <F1> key, and the relevant information will be displayed.

Item	Option	Description
Exit Saving changes	Yes NO	Exit System Setup and save your changes.
Save change without Exit	Yes NO	Save Your changes and without exiting system.
Exit Discarding changes	Yes NO	Exit system setup and without saving your changes.
Load Optimal Defaults	Yes NO	Load default values for all SETUP item.
Discarding changes	Yes NO	Load previous values from CMOS for all SETUP items.

4 WATCHDOG, GPIO, AND BYPASS PROGRAMMING

4.1 Watchdog Programming

This section describes the usage of WATCHDOG. AR-B6050 integrated the WATCHDOG that enable user to reset the system after a time-out event. User can use a program to enable the WATCHDOG and program the timer in range of 1~255 second(s)/minute(s). Once user enables the WATCHDOG, the timer will start to count down to zero except trigger the timer by user's program continuously. After zeroize the timer (stop triggering), the WATCHDOG will generate a signal to reset the system. It can be used to prevent system crash or hang up. The WATCHDOG is disabled after reset and should be enabled by user's program.

Intel also provides a Linux watchdog driver to access the feature on AR-B6050. It can be accessed via /dev/watchdog. About the related operations of Linux watchdog, please refer Linux website.

Please refer to the following table to program WATCHDOG properly, and user could test WATCHDOG under 'Debug' program.

Address port: 2E and Data port: 2F	
C:>debug	To enter debug mode.
-o 2E 87 **-o 2E 01** **-o 2E 55** **-o 2E 55**	To enter configuration.
-o 2E 07	To point to Logical Device Number Reg.
-o 2F 07	To select logical device 7 (WATCHDOG).
-o 2E 72 **-o 2F 40**	To select "keyboard reset" as WATCHDOG output to reset system.
-o 2E 72	Preparing to select the unit of timer equals minute or second.
-i 2F	To read the value of index "2F".
-o 2F xx	The value "xx" equals [(value of index "2F") OR (80)]. OR (80): unit is second. OR (00): unit is minute.
-o 2E 73	Preparing to set the WATCHDOG timer value.

-o 2F ##	The value "##" ranges between 01 ~ FF (1 ~ 255 seconds). **00: To disable WATCHDOG.**
-q	To quit debug mode

Notice: The "actual" timer value may not match with the "theoretical". That is because of the tolerance of internal oscillating clock and cannot be adjusted or optimized.

The WATCHDOG sample code of C language as below:

```
//=============================================================================
==
// Rev     Date          Name     Description
//=============================================================================
==
// 1.0     12/16/2009    Willy    W83627EHF WatchDog timer test
//=============================================================================
==

//=============================================================================
==
//   Language include files
//=============================================================================
==
#include <conio.h>
#include <stdlib.h>
#include <stdio.h>
#include <dos.h>

//=============================================================================
==
//   Normal procedure
//=============================================================================
==
void Show_Help();

//=============================================================================
==
```

```
//   Main procedure
//=============================================================================
==
int main(int argc, char *argv[])
{
    unsigned char IO_Port_Address=0x2E;
    unsigned char Time;

    unsigned char Temp;

    if ( argc != 2 )
      {   Show_Help();     return 1;        }

    clrscr();

    Time=atoi(argv[1]);

    // Set Watchdog
    outportb(IO_Port_Address,0x87);      // (EFER) Extended Functions Enable Register
    outportb(IO_Port_Address,0x87);

    outportb(IO_Port_Address,0x2D);      // Point to Global Reg.
                          // Select Multi-Function pin, (Bit0=0 Watchdog Function)
    outportb(IO_Port_Address+1,(inportb(IO_Port_Address+1)&0xFE));

    outportb(IO_Port_Address,0x07);      // Point to Logical Device Number Reg.
    outportb(IO_Port_Address+1,0x08);  // Select logical device 8, (Watchdog Function)

    outportb(IO_Port_Address,0x30);      // Device Active register
    outportb(IO_Port_Address+1,0x01);

    outportb(IO_Port_Address,0xF5);      // Select Watchdog count mode seconds or minutes
    outportb(IO_Port_Address+1,0x02);       // Default is second and KBRST mode.

    outportb(IO_Port_Address,0xF6);      // Set Watchdog Timer Value
    outportb(IO_Port_Address+1,Time);  // 0x00 to disable, max 0xFF

    textcolor(YELLOW);
```

```
    for(Temp=Time;Temp>0;Temp--)
     {
        outportb(IO_Port_Address,0xF6); // Read Watchdog Timer Value
        Time=inportb(IO_Port_Address+1);

        gotoxy(20,10);
        cprintf(">>> After %3d Second will reset the system. <<<",Time);

        delay(1000);
     }

    textcolor(LIGHTRED);
    gotoxy(18,10);
    cprintf("If you can see this message, Reset system is Fail");

    return 1;
}

//=========================================================================
==
// Function     : Show_Help()
// Input    : -
// Change       : -
// Return : -
// Description    : Show Help string.
//=========================================================================
==
void Show_Help()
{
    clrscr();
    printf("WatchDog Test for W83627EHF\n\n");
    printf("Sample:                                  \n");
    printf("              WDT.EXE 10                 \n");
    printf("( For 10 seconds to reset. )\n");
}
//=========================================================================
==
```

4.2 GPIO Programming

This section describes the usage of GPIOs. AR-B6050 integrated eight bits, 5V TTL level, bidirectional, and software programmable GPIOs for user's application. They are all capable of 5 mA source current for output and 8 mA sink current for input individually. The electrical characteristics of GPIOs as following table:

PIN	SIGNAL	PIN	SIGNAL
1	GPO0	2	VCC
3	GPO7	4	GPI7
5	GPO2	6	GPI6
7	GPO3	8	GPI5
9	GND	10	GPI4

To quickly understand the GPIO programming under Linux, we also provide a sample application source code in product CD, naming gpio.c. It can be used to control GPIO pin described above and also LED.

The GPIO sample code of C language as below:

```
//=============================================================================
==
// Rev     Date            Name     Description
//=============================================================================
==
// 1.0     03/17/10 Willy      GPIO10~GPIO17 Test utility for W83627EHF.
//=============================================================================
==

//=============================================================================
==
// Turbo C++ Version 3.0 Copyright(c) 1990, 1992 by Borland International,Inc.
//=============================================================================
==

//=============================================================================
==
//   Language include files
//=============================================================================
```

```
==
#include <conio.h>
#include <stdio.h>

//=====================================================================
==
//   Normal procedure
//=====================================================================
==
void Show_Help();
void Show_Fail();
void Show_Pass();

//=====================================================================
==
//   Main procedure
//=====================================================================
==
int main(int argc)
{
   char *Model_Name="AR-B6050";
   char *Version="v1.0";
   unsigned char  IO_PORT_BASE=0x2E;   // DATA_PORT = IO_PORT_BASE + 1;
   unsigned char data;
   int result=0;

   if ( argc > 1 )
      {   Show_Help();     return 1;   }

   clrscr();
   textcolor(WHITE);
   gotoxy(1, 1);
cprintf("<>=============================================================
=======<>");
   gotoxy(1, 2);   cprintf("|| W83627EHF GPIO Test Utility %s Acrosser Technology Co., Ltd.
||",Version);
   gotoxy(1, 3);
cprintf("<>=============================================================
```

```
=======<>");
   gotoxy(1, 4);
cprintf("<>==========================================================
=======<>");
   gotoxy(1, 5);   cprintf("|| Model Name    :
||");
   gotoxy(1, 6);   cprintf("|| SIO IO Base :
||");
   gotoxy(1, 7);
cprintf("<>==========================================================
=======<>");

   // Show Got Parameter Informat
   textcolor(LIGHTGRAY);
   gotoxy(18,5);      cprintf("%s",Model_Name);
   gotoxy(18,6);      cprintf("%X",IO_PORT_BASE);

   // Enter W83627EHF Config
   outportb(IO_PORT_BASE,0x87);
   outportb(IO_PORT_BASE,0x87);

   // Set Multi-function Pins to GPIO
   outportb(IO_PORT_BASE,0x29);
   outportb(IO_PORT_BASE+1,(inportb(IO_PORT_BASE+1) | 0x01));

   // Select GPIO Port device
   outportb(IO_PORT_BASE,0x07);
   outportb(IO_PORT_BASE+1,0x07);

   // Set GPIO Port Active
   outportb(IO_PORT_BASE,0x30);
   outportb(IO_PORT_BASE+1,0x01);

   // Set GPIO I/O Register to 00h
   outportb(IO_PORT_BASE,0xF3);
   outportb(IO_PORT_BASE+1,0x00);

   // Set W83627EHF GPIO10~13 to Output, GPIO14~GPIO17 to Input
```

```
outportb(IO_PORT_BASE,0xF0);
outportb(IO_PORT_BASE+1,0xF0);

// Set W83627EHF GPIO10~13 to High
outportb(IO_PORT_BASE,0xF1);
outportb(IO_PORT_BASE+1,0x0F);
// Read W83627EHF GPIO14~17 Status, if not High error.
data=inportb(IO_PORT_BASE+1)&0xF0;
if(data!=0xF0)
    result=1;
// Set W83627EHF GPIO10~13 to Low
outportb(IO_PORT_BASE,0xF1);
outportb(IO_PORT_BASE+1,0x00);
// Read W83627EHF GPIO14~17 Status, if not Low error.
data=inportb(IO_PORT_BASE+1)&0xF0;
if(data!=0x00)
    result=1;

// Set W83627EHF GPIO10~13 to input, GPIO14~GPIO17 to Output
outportb(IO_PORT_BASE,0xF0);
outportb(IO_PORT_BASE+1,0x0F);

// Set W83627EHF GPIO14~17 to High
outportb(IO_PORT_BASE,0xF1);
outportb(IO_PORT_BASE+1,0xF0);
// Read W83627EHF GPIO10~13 Status, if not High error.
data=inportb(IO_PORT_BASE+1)&0x0F;
if(data!=0x0F)
    result=1;
// Set W83627EHF GPIO14~17 to Low
outportb(IO_PORT_BASE,0xF1);
outportb(IO_PORT_BASE+1,0x00);
// Read W83627EHF GPIO14~17 Status, if not Low error.
data=inportb(IO_PORT_BASE+1)&0x0F;
if(data!=0x00)
    result=1;

// Exit W83627EHF Config
```

```
    outportb(IO_PORT_BASE,0xAA);

    if(result)
        Show_Fail();
    else
        Show_Pass();

    return result;
}

//=====================================================================
==
// Function     : Show_Help()
// Input    : -
// Change       : -
// Return : -
// Description    : Show Title string.
//=====================================================================
==
void Show_Help()
{
     clrscr();
     printf("GPIO Test utility for W83627EHF\n\n");
     printf("GPIO0 ▯◀迂迂迂苇          ▯ Vcc\n");
     printf("GPIO1 ▯◀迂迂迂銀迂迂┼▯ GPIO7\n");
     printf("GPIO2 ▯◀迂迂苇    迂迂┼▯ GPIO6\n");
     printf("GPIO3 ▯◀迂迂銀迂迂迂┼▯ GPIO5\n");
     printf("GND     ▯              迂迂迂┼▯ GPIO4\n");
}

//=====================================================================
==
// Function     : Show_Fail()
// Input    : -
// Change       : -
// Return : -
// Description    : Show Fail Message.
//=====================================================================
```

```
==
void Show_Fail()
{
    textcolor(LIGHTRED);
    gotoxy(20,10);    cprintf(" 訶訶訶訶  訶訶訶訶    訶訶訶  訶訶訶訶");
    gotoxy(20,11);    cprintf(" 訶訶訶訶  訶訶訶訶    訶訶訶  訶訶訶訶");
    gotoxy(20,12);    cprintf(" 訶訶訶訶  訶訶訶訶    訶訶訶  訶訶訶訶");
    gotoxy(20,13);    cprintf(" 訶訶訶訶  訶訶訶訶    訶訶訶  訶訶訶訶");
    gotoxy(20,14);    cprintf(" 訶訶訶訶  訶訶訶訶    訶訶訶  訶訶訶訶");
}

//=====================================================================
==
// Function    : Show_Pass()
// Input   : -
// Change     : -
// Return : -
// Description   : Show Pass Message.
//=====================================================================
==
void Show_Pass()
{
    textcolor(LIGHTGREEN);
    gotoxy(20,10);    cprintf(" 訶訶訶訶  訶訶訶訶  訶訶訶訶訶 訶訶訶訶訶");
    gotoxy(20,11);    cprintf(" 訶訶訶訶  訶訶訶訶  訶訶訶訶訶 訶訶訶訶訶");
    gotoxy(20,12);    cprintf(" 訶訶訶訶  訶訶訶訶  訶訶訶訶訶 訶訶訶訶訶");
    gotoxy(20,13);    cprintf(" 訶訶訶訶  訶訶訶訶  訶訶訶訶訶 訶訶訶訶訶");
    gotoxy(20,14);    cprintf(" 訶訶訶訶  訶訶訶訶  訶訶訶訶訶 訶訶訶訶訶");
}
//=====================================================================
==
```